



Procedia Computer Science

Volume 41, 2014, Pages 146–151

BICA 2014. 5th Annual International Conference on Biologically  
Inspired Cognitive Architectures

# The Equilibration of Neural Propositions

**Michael S P Miller***Piaget Modeler  
Los Angeles, CA, USA  
piagetmodeler@gmail.com***Sheldon O Linker***Linker Systems, Inc.  
Irvine, CA, USA  
sol@linker.com***Ghassan Azar***Lawrence Technological University  
Southfield, MI, USA  
gazar@ltu.edu*

## Abstract

Jean Piaget explained equilibration as the process by which individuals create cognitive structures to compensate for disturbances encountered during their interactions with the world. Using Neural Propositions as its basic unit of representation, the Piagetian Autonomous Modeler (PAM.P2) is a cognitive system that employs equilibration to refine its world model as it compensates for disturbances encountered in its environment. Equilibration is discussed and contrasted with other learning approaches. PAM.P2's implementation of equilibration is reviewed.

**Keywords:** Piaget, PAM.P2, equilibration, neural proposition, cognitive system, reflection, equilibrium, regulation, compensation, failure driven learning, dynamic planning, reinforcement learning

## 1 Introduction

Finding a general theory that a cognitive system can employ to learn from and interact successfully with an environment is a very challenging problem. In his books *The Development of Thought* and *The Equilibration of Cognitive Structures*, Jean Piaget (1977; 1985) proposed the theory of equilibration (compensating for disturbances) as the central means by which people learn from and interact with the world. A cognitive system is in equilibrium with its environment when it has accurate predictions and adequate responses to environmental stimuli. When responses are inadequate, or predictions fail because of knowledge gaps or unforeseen obstacles, the system falls out of equilibrium, and must constructively reorganize itself and reestablish a higher trajectory of equilibrium using newly created structures. Otherwise the system's interactions will continue to fail, and eventually cease.

Finding a general theory for a cognitive system to employ is important for a number of reasons. First, having no theory leads to ad-hoc systems with no clear organization or justification for components, integrations, or processes. Theory driven systems often have better architectural design and organization, and better justification for components, integrations, processes and data structures than ad-hoc systems. Second, using highly specialized theories leads to brittle and narrow situation specific systems; using general theories leads to more resilient and robust systems with broad scope

and applicability. Third, agglomerating a few specific theories fails to provide the cohesion that comes with a general theory, and may lead to systems constructed with gaps.

Equilibration is a general theory that is independent of any specific implementation. It is a constructivist theory and as such, it favors cognitive systems that have elements which can be combined or separated as the primary means of self-improvement, and favors approaches that qualitatively modify or synthesize elements which augment the extension, domain, and range of the representations in their models.

Systems having elements such as rules (Laird, Rosenbloom and Newell [1987]) or cases (Schank [1982]; Hammond [1989]) are good candidates for equilibration if the proper processes for element adaptation, combination and separation can be implemented. Meta, Ontañón and Ram (2009) use failure patterns to modify failed plans in a real time strategy case based planning system. The system ignores some disturbances (plan failures) while incorporating and transforming others.

Equilibration has some commonality with failure driven learning since both seek to exploit the failures of a system rather than the successes. Stroulia (1994) has explored agents modifying their own architecture based on an internal self-model in the face of failures. Whereas failure driven learning typically deals with systems disconnected from an environment, equilibration deals with systems continuously immersed in an environment. Furthermore, equilibration assumes that the processes controlling a cognitive system are fixed and not subject to variation, while the data structures containing the experiences of the system are modifiable. Others (Cox and Ram [1994]) have looked at modifying input parameters to a fixed problem solving system based on encountered failure episodes. Equilibration assumes the input to the cognitive system can only be controlled to a limited extent by the actions of the cognitive system rather than through direct manipulation of training instances.

Dynamic planning systems which compute the next action for a given context by integrating planning and execution bear close resemblance to equilibrative systems. Chien et al. (1999) propose an iterative repair problem solver which continually adapts an operations plan for an autonomous spacecraft. Their system maintains a current goal set, current state, and forward state. Modifying the goals or current state updates the plan and forward state. Disturbances arise when current actions cannot be performed due to constraint violations on the current state or when required resources are unavailable. In light of disturbances, replanning is performed to adapt the plan to the current state. This system demonstrates compensation as it circumvents or incorporates disturbances. In contrast to equilibrative systems, Chien et al.'s system's world model is fixed and any new constructions do not extend the domain or range of the world model, thereby limiting any predictive ability the system may have. Also, the conditions under which actions can be performed are predetermined and fixed, rather than discovered by the system. This stands in contrast to Drescher's (1991) schema mechanism which learns the conditions under which actions can be successfully performed. In an equilibrative system the conditions under which actions succeed are expected to be learned by the system. To the extent that they rely on fixed world models which cannot be constructively augmented, or rely on prespecified constraints for actions that are not learned, or have disturbances which are known a priori, dynamic planning systems fall short of being equilibrative systems. A synopsis of Piaget's theory and description of how it is implemented in the PAM.P2 (Miller [2013]) cognitive system follows.

## 2 Equilibration

According to Piaget, to understand equilibration we must first talk about systems and equilibrium. Biological systems are both open and closed. Biological systems require external elements (such as food, water, or light) in order to function, so biological systems are open insofar as they exchange elements with the environment, and are closed because these exchanges form closed cycles of interaction between the system and its environment. Like biological systems, cognitive systems are

also both open and closed; however, unlike biological systems, cognitive systems do not need external elements to function: they can operate in a closed manner using only internal elements: elements of thought. The internal elements of a cognitive system can be activated without recourse to external elements and the cognitive system can operate using them.

Biological and cognitive systems form cycles of interaction with their environments (milieus). External elements in the environment can be incorporated (assimilated) into a system. For cognitive systems these external elements correspond to stimuli, sensations, perceptions, or features, collectively called observables. Internal elements within the system (called coordinations) can be activated either by the external elements when they are perceived, or by other internal elements. As these activations chain, they form a cycle of interaction between the system and the environment. When the cycles are stable and predictable, the system is said to be in equilibrium with the environment. When a cycle is broken because of missing or newly introduced external elements, the cycles are disturbed. The system must adapt to the disturbance.

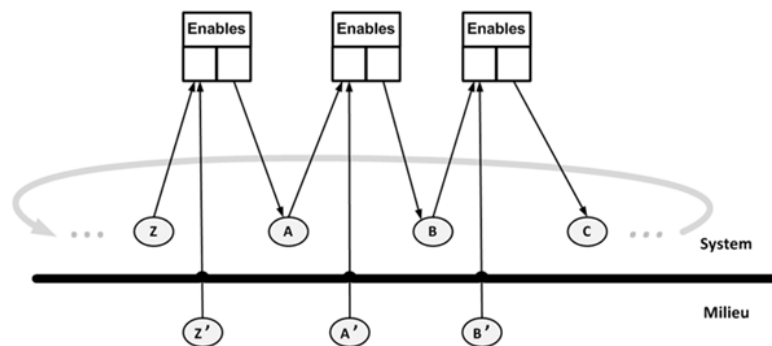


Figure 1. A stable cycle. Patterns in the milieu perceived by sensors are interpreted as external elements  $A'$ ,  $B'$ , ...  $Z'$ . Schemes are created (e.g., **Enables**( $x,y$ ) ) to incorporate the external and internal elements  $A$ ,  $B$ ,  $C$  ...  $Z$ , forming a cycle of interaction (and activation).

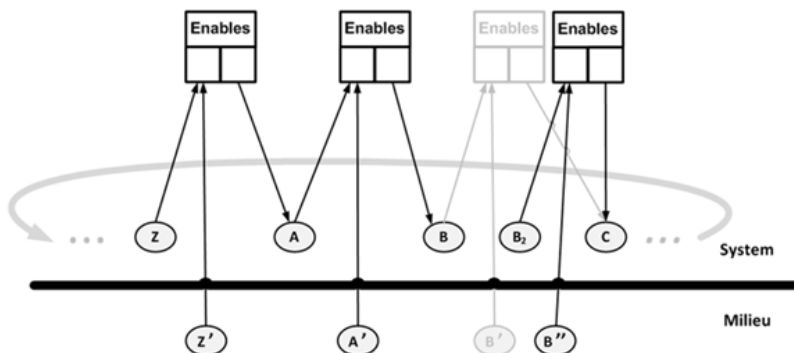


Figure 2. When external elements (e.g.  $B'$ ) do not recur or new elements (e.g.  $B''$ ) are introduced, the system may reject the changes and terminate the cycle, or adapt to the changes by incorporating the new elements (e.g.,  $B''$ ) and by “accommodating” portions of itself (e.g., synthesizing element  $B_2$  and cloning a new scheme, **Enables**( $\{B_2, B''\}, \{C\}$ ) ) to form augmented interaction cycles.

The incorporation of external elements into a system is called assimilation. Accommodation means changing the system itself to adjust to the external element. Through assimilation and accommodation a system can adapt itself to handle the disturbances it encounters in its environment.

### 3 Disturbances and Compensations

Constructive reorganization is the basis of Piaget’s genetic epistemology: “knowledge proceeds neither solely from the experience of objects nor from an innate programming performed in the subject but from successive constructions, the result of the constant development of new structures...leading...to reequilibrations which improve the previous structures” (Piaget [1977], p. v).

A cognitive system such as PAM.P2 is in equilibrium with its environment when events can be anticipated, urges can be satisfied, and goals can be pursued and achieved. Disturbances which make events unpredictable, thwart goals, or leave urges unsatisfied, cause disequilibrium. Disturbances typically come in two varieties, unforeseen obstacles (impediments, failure reasons, errors, exceptions) and gaps (missing knowledge needed for problem solving). Disturbances are compensated for by regulative and compensatory processes which use assimilation and accommodation. In the PAM.P2 cognitive system assimilation means adding elements to the system and accommodation means cloning and modifying elements.

When a disturbance arises, a cognitive system has three choices: (1) ignore or reject the disturbance thereby circumventing it, (2) incorporate the disturbance, or (3) transform the disturbance. In the case of a solution failure during planning, a cognitive system can simply choose another solution to try. Selecting a different solution ignores the reasons why the first solution failed, and misses the opportunity to gain critical differentiating knowledge. Here there is a tradeoff between learning and problem solving. When a prediction fails, a cognitive system can simply check other predictions. In both cases the system can choose to ignore the disturbance. A system can also incorporate a disturbance into itself. For example, the situation that caused a failure can be saved in episodic memory so if the situation recurs it can be expeditiously utilized as control knowledge. If a prediction fails, the system can regulate the prediction by demoting its reliability or utility scores, and then compensate for the failure by adding an ascribed failure cause to a clone of the prediction. In both cases the disturbance becomes an expected exception in the normal interaction flow. Finally a cognitive system can transform a disturbance. The system can set goals to cancel the disturbance or neutralize its unwanted effects; or, the system can create inverse solutions to return the situation to a pre-disturbance state. Thus, disturbances can be circumvented, incorporated, or transformed.

### 4 PAM.P2

The PAM.P2 cognitive system (Miller [2013]) initially receives a message from a device indicating that the device is working and available to establish a session; possible actions and device needs are conveyed in the message. Thereafter the device proceeds to send sensorimotor datasets, urges, and action results to PAM.P2. PAM.P2 continually sends action requests to the device.

The perceiver agent within PAM.P2 receives all messages from the device and stores them as propositions in PAM.P2’s memory. An attention agent converts urge messages into intentions to satisfy the associated needs. Activation agents close satisfied intentions. Deliberative and reactive action selector agents find applicable solutions to satisfy intentions on the agenda, and create forward situations and pending trials of the solution. Solver agents propose solutions if none exist. An executor agent looks for selected solution trials, creates subgoals for disabled preconditions, and decomposes the solutions into action requests which it sends to the device. When action results from the device are returned, correlator agents update the solution trials, calculate system affect, mood and equilibrium values, and appraise open intentions. Regulator agents review trials and correct or reinforce predictions or solutions. A compensator agent looks for corrected solutions (see Table 2). The attention agent compensates for (i.e., copes with) disturbing appraisals by making modifications to intentions on the agenda.

No.	Disturbance Type: Example
1.	<u>Failed Solution Trial:</u> [Try ^ Try_971 :Solution Solution_885 :Start 2014660932264491m :Until 2014660933811199m :Outcome failure] [Solution ^ Solution_885 :Goal Monad_253 :Before {Monad_45} :Always {} :Never {Monad_562} :Steps {Action_2 Action_2 Action_1} :After {Monad_253 Monad_7772} :Timeframe 9244m :Inverse nothing]
2.	<u>Failed Prediction:</u> [Prediction ^ Prediction_809 :Hypothesis Hypothesis_75 :Start 2014660932264491m :Until 2014660933820443m :Outcome failure] [Hypothesis ^ Hypothesis_75 :Type empirical :Before {Monad_838} :After {Monad_712} :Timeframe 52449m]
3.	<u>Failed Action Attempt:</u> [Result ^ Result_454 :Attempt Attempt_43 :Outcome failure] [Attempt ^ Attempt_43 :Action Action_2 :Solutions {Solution_885} :Start 2014660863158992m :Until 2014660865618214m :Outcome failure]
4.	<u>Unsatisfied Urge:</u> [Urge ^ Urge_2 :Need Need_2 :Urgency 87] [Intention ^ Intention_567 :Goal Monad_998 :Start 2014660932264491m :Until infinity :Urgency 87 :Salience 100 :Status open]

Table 1. Sample disturbances in PAM.P2 - colons precede slot names, ^ is the identifier slot.

```
(namespace pam.p2.Mind
```

```

(prototype SolutionCompensatorAgent uses Agent
!job SolutionCompensatorAgent.compensate )

(function SolutionCompensatorAgent.compensate
(given [Try ^ ?try :Solution ?solution (= @:Outcome regulated)]
[Solution (= @^ ?solution) :Inverse ?inverse]
do
; if the solution doesn't have an inverse then create one
(if (nil ?inverse)
(createInverseSolution ?solution))
; get unanticipated effects of the trial and neutralize them
(for ?effect in (getUndesiredSideEffects ?try)
(set (knew Intention :Goal (non ?effect)) :Start (moment))))))
)
```

Table 2. The solution compensation agent in PAM.P2 (written in the Premise language).

PAM.P2 performs regulation by (1) increasing the utility and reliability of successful solutions and predictions (2) recombining successful solutions and predictions (3) decreasing the utility and reliability of failed solutions and predictions (4) cloning and mutating failed solutions and predictions to include contemporaneous preconditions, and (5) ascribing failure causes (i.e., problems). PAM.P2 achieves solution compensation in two ways, by constructing new solutions with reversed actions and by creating intentions to eliminate undesired post conditions. PAM.P2 compensates for appraisals by (1) creating, (2) reprioritizing, (3) suspending, (4) resuming, or (5) abandoning intentions.

## 5 Neural Propositions

Some propositions in PAM.P2 are neural propositions (Miller [2013]), meaning that they have an extended network of support propositions facilitating activation and belief propagation. Each neural

proposition belongs to one or more reasoning contexts, called viewpoints. Within a viewpoint a proposition can be activated along three dimensions: fact, goal, and neme. Fact activation is based on present observations of the cognitive system. Goal activation is based on what the cognitive system wants to achieve at a given point in time. Following Minsky (Singh [2003]), a neme represents an activation in a past problem solving episode, facilitating activation through reminding. Each neural proposition has an affirmative aspect, called a “thesis monad,” and a negative aspect, called an “antithesis monad.” When a neural proposition is enabled, the thesis monad is enabled while the antithesis monad is impeded. Both monads may be deactivated through decay over time, or by being simultaneously voided. Activation occurs in PAM.P2 by stamping the current moment on the appropriate activation instance for a given monad, dimension, viewpoint combination (Miller [2013]).

## 6 Discussion and Future Work

The PAM.P2 system is currently in development and test. Testing developmental cognitive systems is challenging because there is no specific task that must be accomplished to show progress. Rather, the adaptation of the system must be measured over time. Guerin and Rat-Fischer (2014) address the issues of testing developmental systems. The notion of equilibration offers Artificial Intelligence practitioners another perspective upon which to design, measure and build systems.

## References

- Chien, S., Knight, R., Stechert, A., Sherwood, R., Rabideau, G., (1999) Using Iterative Repair to to Increase the Responsiveness of Planning and Scheduling for Autonomous Spacecraft in proceedings of International Joint Conference on Artificial Intelligence, IJCAI 1999, Stockholm Sweden, August 1999
- Cox, M. T., Ram, A. (1994) Failure-Driven Learning as Input Bias in proceedings of the Sixteenth Annual Conference of the Cognitive Science Society, Hillsdale, NJ, Lawrence, Erlbaum
- Drescher, G. (1991) *Made Up Minds: A Constructivist Approach to Artificial Intelligence*. MIT Press
- Guerin, F., Rat-Fischer, L. (2014) *Benchmarking in Developmental Robotics*. Unpublished.
- Hammond, K.J. (1989), *Case Based Planning: Viewing Planning as a Memory Task*, Academic Press
- Laird, J., Rosenbloom, P., Newell, A. (1987). "Soar: An Architecture for General Intelligence". *Artificial Intelligence*, 33: 1-64.
- Mehta, M., Ontañón, S., and Ram, A. (2009). Using Meta-Reasoning to Improve the Performance of Case-Based Planning in International Conference on Case-Based Reasoning, ICCBR 2009
- Miller, M. S. P. (2013) The Neural Proposition: Structures for Cognitive Systems, in the proceedings of the AAAI 2014 Spring Symposium on Creativity and (Early) Cognitive Development, Stanford, CA USA
- Piaget, J., Rosin, A. (1977) *The Development of Thought: Equilibration of Cognitive Structures*, Viking Press
- Piaget, J., Brown, T., Thampy, K.J. (1985) *The Equilibration of Cognitive Structures*, The University of Chicago Press
- Schank, R. C. (1982). *Dynamic memory: A theory of reminding and learning in computers and people*. Cambridge University Press
- EasyChair conference system*. Retrieved from easychair.org
- Singh, P. (2003) Examining the Society of Mind in Computing and Informatics Vol 22, p. 521-543
- Stroulia, E. (1994) Failure-Driven Learning as Model-based Self Redesign. Georgia Institute of Technology. GIT-CC-95-38